

R-Matrix  
\*\*\*\*\*

These Classic R-matrix and related codes do not correspond exactly to any published or unpublished R-matrix codes. However, they do contain features not found in the Belfast/UCL codes which you may find of use. The Belfast (inner-region) codes have been published in Computer Physics Communications (CPC v92 p290 1995 - BP/LS exchange and v62 p76 1992 - LS non-exchange) and are available via the WWW from <http://www.cpc.cs.qub.ac.uk/cpc/> under catalogue numbers ADCP and ACGP. The UCL/Belfast (outer region) codes remain unpublished.

WARNING: DO NOT MIX these R-matrix codes with any others!

N. R. Badnell 06/05/09

oo

INNER REGION CODES  
\*\*\*\*\*

Familiarity at least with the latest (non-CPC) NAMELIST/free-format input, and operation, would be helpful but it is not essential.

To use the local executables you should add /home/badnell/rmatrix/x to your path - and have an account on a local machine!

Prior to running any R-matrix codes you should have generated previously a 'radial' file (called 'radout' by AUTOSTRUCTURE) that will be used by R-matrix to re-generate your previously determined atomic structure. If you haven't done this then go back and read the AUTOSTRUCTURE file WRITEUP.

A suitable dataset for C2+ is:

```
123456789 21522 21512513 21523
          10 20 21
&SALGEB RUN=' ' RAD='NO' CUP='LS' TITLE='C2+ test' KORB1=0 KORB2=0 &END
&SMINIM NZION=6 INCLUD=0 PRINT='FORM' NLAM=3 NVAR=0 RADOUT='YES' &END
1.3678 1.2308 1.1670
```

Then copy or move the file 'radout' to 'radial'.

The (local) full exchange executables for non-relativistic (see 1. here) and Breit-Pauli (see 2. below) operation are:

```
stglr.x stg2r.x (stgjk1.x) stg3r.x
```

on SUNs, replace .x with .xsg on the SG machines.

1. Non-relativistic (LS-coupling) operation.

-----

stgjk1.x is omitted during an LS run (this is the RECUPD code).

The file containing the UNIT5 input detailed below is specified by the redirect "<", e.g. stglr.x <dstg1 and the UNIT6 output automatically goes to a file called 'routnr' where n will be 1,2,3 etc.

```
STG1: read bound orbitals (from file 'radial'), generate continuum basis
**** orbitals and evaluate all possible bound-bound, bound-continuum and
continuum-continuum integrals that could be required.
```

## STG1 Input

-----

The first four characters of line 1 MUST BE: S.S. unless you know better.

Then the namelist STG1A input:

None normally, but

ISMITN = 1 Use new orthogonalization method with pseudo-state expansion.  
 = 0 Default, use original recurrence relation (unstable).

Then the namelist STG1B input:

MAXLA = maximum N-electron (atomic) angular momentum.

MAXLT = maximum (N+1)-electron (total) angular momentum.

MAXC = maximum number of continuum basis orbitals per angular momentum.

MAXE (optional) if specified, is the maximum scattering energy in Rydbergs for your problem. If the MAXC you have specified is too small, an estimate of the value required is printed and execution halts.

Note, for a given MAXC, the maximum scattering energy for which accurate collision strengths can be calculated is one half the maximum eigen-energy of the continuum basis orbitals (approximately).

ISMIT(I)=nl, denotes which, if any, of your nl orbitals are non-spectroscopic.

e.g. ISMIT(1)=40 ISMIT(2)=41 labels the 4s and 4p orbitals as pseudo.

Alternatively, use NMIN, NMAX, LMIN, LMAX to give a range of nl-values.

Or, if you set NMAX.LT.NMIN then specify NPS(l)=n where l=pseudo a.m. and n is the max (pseudo) principal quantum number for that l. The min is given by NMIN of course.

Example: C2+

S.S. C2+ TARGET 1 CPC TEST

&STG1A &END

&STG1B MAXLA=2 MAXLT=8 MAXC=25 MAXE=4 &END

++++  
 To use STO orbital specification, instead of a radial file but everything else as above, then replace S.S. by STO- and set NZED, NELC, MAXORB in STG1B with MAXORB as per STG2 (next).  
 +++++

STG2: Evaluate N- and (N+1)-electron angular algebra. Form N- and (N+1)-electron  
 \*\*\*\* Hamiltonians. Diagonalize N-electron Hamiltonian to regenerate the target eigen-energies (and eigen-vectors).

## STG2 Input

-----

The first four characters of line 1 MUST BE: S.S. unless you know better.

Then the namelist STG2A input:

None, normally.

Then the namelist STG2B input:

MAXORB = maximum number of orbitals that will be used to define the target configurations. Their nl values follow the namelist.

NELC = number of target electrons.

NAST = number of target terms, specified as 2S+1 L p after the namelists (S.S.),  
 specified as L 2S+1 p after the namelists (STO-).

INAST = number of total SLp symmetries, specified as 2S+1 L p after the  
 namelist. Set p=0 for even parity and p=1 for odd parity.

If INAST = 0 then

MINLT = minimum total L.

MAXLT = maximum total L.

MINST = minimum total 2S+1.

MAXST = maximum total 2S+1.

SLp's will be generated internally for this range, both parities.

endif

The target configurations are specified via orbital occupation numbers.  
 It is possible to allow promotions from a basic configuration subject  
 to an overall minimum and maximum occupation constraint.

The (N+1)-electron configurations, which are added to compensate for the  
 enforced orthogonality of the continuum orbitals to the bound orbitals,  
 are specified in a similar manner.

You must specify at least those (N+1)-electron configurations that arise from  
 adding any of the bound orbitals to an N-electron configuration that gives  
 rise to a term that is specified by NAST.

If you specify (N+1)-electron configurations that arise from adding a bound  
 orbital to a configuration that includes a term not specified by NAST then  
 you may run into problems with pseudo-resonances (no details!).

The simplest, safest solution for the non-expert is to use N-electron  
 configurations that are purely CC (i.e. all terms are specified by NAST) or  
 purely CI (i.e. containing no terms specified by NAST). You then use the  
 MAXORB orbitals to form all possible (N+1)-electron configurations from the CC  
 N-electron configurations BUT NOT from the CI configurations.

Now would seem like a good time for an example: C2+ again

S.S. C2+ TARGET 1 CPC TEST

&STG2A &END

&STG2B MAXORB=3 NELC=4 NAST=6 INAST=0 MINLT=0 MAXLT=8 MINST=2 MAXST=4 &END

1 0 2 0 2 1

3

2 0 0

2 2 2

2 2 0 0

2 1 1 0

2 0 2 0

1 0 0

1 0 0

3 1 1

1 1 1

3 1 0

1 2 0

3

2 0 1

2 2 3

2 2 1 0

2 1 2 0

2 0 3 0

Note, the input is all free-formatted.

a) Following the namelists, the three nl orbitals are specified.

b) Then the number of target configurations is specified, 3.

c) Then the minimum and maximum occupation numbers of the target configurations are specified, these constrain possible configurations allowed by promotions from a basic configuration.

d) Then the 3 target configurations themselves are specified, the final zero denotes that no promotions are allowed from these 3 basic configurations, they are the only configurations required.

e) The NAST=6 target terms are then specified. While it is natural to list them by energy order it is in fact computationally much more efficient to group them by symmetries, as has been done for the singlet S even, so that the angular algebra is solved-for only once per target symmetry.

f) Then b), c) & d) are repeated, this time for the (N+1)-electron configurations. If INAST were non-zero then e) would also be repeated.

STG3 (sometimes known as STGH): Diagonalize (N+1)-electron Hamiltonian.  
\*\*\*\*\*

STG3 Input

-----  
The first four characters of line 1 MUST BE: S.S. unless you know better.

Then the namelist STG3A input:

None, normally.

Then the namelist STG3B input:

INAST = 0 (default) then all (N+1)-electron symmetries specified in STG2 are looped-over.

> 0 then specify the SLp's to be used, this is a much slower operation and should only be used if really needed.

NAST = 0 (default) do nothing.

> 0 Read NAST experimental energies in Rydbergs relative to the ground. Adjustments are then made to the diagonal of the continuum-continuum part of the Hamiltonian.

Example: C2+ yet again

S.S. C2+ TARGET 1 CPC TEST

&STG3A &END

&STG3B INAST=0 NAST=6 &END

0.0000 1.6632 0.4775 0.9327 1.2526 1.3293

Note, the energies are free-formatted and are input in the order that the terms were listed in STG2 (after any re-ordering), or the levels were listed in recupd/stgjk.

-----

## 2. Breit-Pauli operation

-----  
This is not too different from the non-relativistic operation.  
Remember to execute stgjk1.x (aka RECUPD) after stg2r.x.

In addition to the input described for non-relativistic operation, for

STG1:

\*\*\*\*

Add to namelist STG1A, RELOP='YES'

STG2:

\*\*\*\*

Add to namelist STG2A, RELOP='YES'

Note, the N-electron Hamiltonian is not diagonalized now and the CC expansion is forced (expanded) to be the same as the CI expansion.

Also, ensure MAXLT is set large enough (.GE.J+MAX(S)) so that all possible matrix elements required for subsequent recoupling are present.

STGJK: (aka RECUPD) Recouple N- and (N+1)-electron Hamiltonians. Diagonalize  
\*\*\*\*\* N-electron Hamiltonian to regenerate the target eigen-energies (and eigen-vectors).

STGJK Input

-----  
The first four characters of line 1 MUST BE: S.S. unless you know better.

Then the namelist STGJA input:

None, normally.

Then the namelist STGJB input:

JNAST = number of target levels, specified as 2J p after the namelist.

IJNAST = number of total Jp symmetries, specified as 2J p after the namelist.

If IJNAST = 0 then

J2MIN = minimum total 2\*J.

J2MAX = maximum total 2\*J.

Jp's will be generated internally for this range, both parities.

endif

Example: C2+ ad nauseum

S.S. C2+ TARGET 1 CPC TEST

&STGJA &END

&STGJB JNAST=10 IJNAST=0 J2MIN=1 J2MAX=5 &END

0 0 1S

0 1 3P

2 1 3P

4 1 3P

2 1 1P

0 0 3P

2 0 3P

4 0 3P

4 0 1D

0 0 1S

Note, the "term" label after the target levels is for identification purposes only.

(Aside, is there any efficiency in grouping like levels together?)

STG3 and the outer region codes now follow as for non-relativistic operation except that we are dealing with levels rather than terms. Any reference to L is replaced by 2J and any reference to 2S+1 L p by 0 2J p.

In addition, TERM COUPLING COEFFICIENTS can be obtained as well by setting RELOP='TCC' in STG2A, they are actually output to TCCDW.DAT by STGJK. In this instance STG2 evaluates the N-electron Hamiltonian twice. Firstly to evaluate the LS e-energies and e-vectors, INCLUDING Mass-Velocity and Darwin terms. So to make a meaningful comparison between TCC and LS results the LS results should include Mass-Velocity and Darwin terms. This can be achieved by running STG1 with RELOP='YES' or 'MVD' and STG2 with RELOP='MVD', then STG3 etc. as in a non-relativistic run. STGJK is ONLY needed when the spin-orbit interaction has been switched-on via RELOP='YES' or 'TCC' in STG2. Secondly, STG2 evaluates the N-electron B.P. Hamiltonian including the spin-orbit interaction ready for recoupling and diagonalization by STGJK. The TCCs are in addition to a full Breit-Pauli run. If only TCCs are required, e.g. from a large LS calculation, then set RELOP='TCC' and IDWOUT=2 in stg1r/stg2r/stgjk for maximum efficiency (stgjk=recupd).

### 3. Photoionization operation

The non-relativistic operation follows 1., and Breit-Pauli follows 2. EXCEPT that one MUST use the local executable stgjk2.x, which has the photoionization memory switched-on (MZIPH=2).

In addition to the input described in 1. and 2., for

STG1:  
\*\*\*\*\*

Add to namelist STG1A, RAD='YES'

STG2:  
\*\*\*\*\*

Add to namelist STG2A, RAD='YES'

(In Breit-Pauli mode the (N+1)-e symmetries MUST be grouped by parity, even then odd.)

STGJK: (For level-level)  
\*\*\*\*\*

Add to namelist STGJA, RAD='YES'

Also, the target levels MUST be grouped by parity.

STG3:  
\*\*\*\*\*

Add to namelist STG3A, RAD='YES'

Note, STG3 generates the dipole matrix elements in the files Dnm.

The outer region codes STGB and STGF should be run next, generating the Bnm and Fnm files for the bound and free matrix elements. Finally, PREBF and STGBF generate the photoionization cross sections. See the outer region code descriptions below.

---

#### 4. Non-exchange (non-relativistic) code operation

---

The non-exchange (non-relativistic ONLY) executables are (see 4. here)

stg1nx.x stg2nx.x stg3nx.x

The 3 stages are:

STG1NX (aka NXANG): which calculates the angular algebra.

\*\*\*\*\*

STG2NX (aka NXRAD): which calculates the basis orbitals and radial integrals.

\*\*\*\*\*

STG3NX (aka NXHAM): which diagonalizes the N- and (N+1)-electron Hamiltonians.

\*\*\*\*\*

We run the NX codes as a continuation of the exchange codes. Thus, the files named NX1.DAT and NX2.DAT, generated automatically during an exchange run, must be present. They are used by all 3 stages. All 3 stages read the unit5 file called 'dstgnx' which is of the form:

CONTinuation run.

```
&STGNX  MINLT=9  MAXLT=12  &END
```

The first 4 characters of line 1 must be CONT.

Then,

MINLT = minimum total angular momentum (generally = exchange code MAXLT+1).

MAXLT = maximum total angular momentum (I have tested up to 80).

In addition, the (exchange) STG3 dataset must also be present and named 'dstg3'. Only the observed energies are read from this file. If only calculated energies are used then only a generic file needs to be present.

The UNIT6 output files are 'routmnx' with m=1,2,3.

A standard H.DAT file is produced for use by an outer-region code (see below).

oo

#### OUTER REGION CODES

\*\*\*\*\*

FREE

\*\*\*\*

They read the H.DAT file generated by STG3(NX) and calculate collision strengths at a user supplied set of energies and write them to an OMEGA file. Note, all input energies are z\*\*2 scaled in the residual charge, if z>0.

oooooooooooooooooooooooo

STGF: Seaton's perturbative treatment of the long-range coupling potentials.

\*\*\*\* Now extended to handle neutrals (NRB).

Local executable stgf.x

STGF Input

-----

## Namelist STGF

IQDT = 0 Default, standard STGF operation.  
= 1 MQDT operation via the unphysical S-matrix.  
= 2 MQDT operation via the unphysical K-matrix.

IMODE= 0 Write unphysical K/S-matrix to JBIN file. Default.  
= 1 Read unphysical K/S-matrix from JBIN file.  
=-1 Single pass interpolation of coarse mesh (see IEQ) onto  
the "usual" fine mesh.

IPRKM= 0 Do not write unphysical K/S-matrix to file k/smtls.dat. Default.  
= 4 Write unphysical K/S-matrix to file k/smtls.dat for stgicf.  
By default, these files are split by symmetry viz. k/smtls001,  
k/smtls002, k/smtls003 etc.

IMESH defines the type of energy mesh and subsequent namelist to be read.  
= 1 Constant spacing in energy dE.  
= 2 Constant spacing in effective quantum number dn.  
= 3 An arbitrary set of user-supplied energies.

IEQ controls how often the MQDT K-/S-matrix is calculated.  
= -1 The MQDT K-/S-matrix is calculated at every energy (default).  
> 0 then calculated at IEQ linearly spaced energies across the  
energy range defined by MESH. This is single pass operation.  
(see note: below)

PERT = 'NO' Neglect long-range coupling potentials (fast operation).  
= 'YES' Include long-range coupling potentials (factor 5-10 slower).

LRGLAM >=0 Maximum L-value (or 2\*J) from STG3, partial wave sum will be  
topped-up with an estimate for LRGLAM to infinity.  
< 0 (default) No top-up.

IBIGE = 0 default, does nothing.  
= 1 append infinite energy scaled collision strength to the file OMEGA.

IPRINT defines the print level, -2 is lowest level and +3 the highest.

IRAD = 0 (default) collision strengths only.  
= 1 write the Fnm files for photoionization, as well.  
= 2 ditto, but no computation of collision strengths.

IOPT1 = 1 (default) all SLp cases.  
= 2 specify cases, (2S+1) L p or 0 2J p after energies (see next).  
Terminate with -1 -1 -1.

## Namelist MESH1

MXE = the number of energy mesh points.  
E0 = the first (scaled) energy point.  
EINCR = the (scaled) energy increment.

or

## Namelist MESH2

DQN = effective quantum number step  
QNMAX = maximum effective quantum number of resonances to be resolved.



EMIN = minimum total energy.

EMAX = maximum total energy.

DEOPEN = interval in (EMIN,EMAX) for all channels open.

or

Namelist MESH3

MXE = the number of user supplied (scaled) energies in Rydbergs, relative to the ground state, which are read free-formatted after this namelist.

Note, I tend to use MESH1, with a "small" EINCR through the resonance region  
 \*\*\*\* (~5000 points) and a "large" EINCR where all channels are open (~10 points). But, in MQDT mode (IQDT=1 or 2) the following approach should be used (with IEQ=-1, default):

Run with a coarse energy mesh first (IMODE=0, default).

Then re-run with a fine energy mesh (set IMODE=1).

For production runs a single pass (IMODE=-1) is more efficient, making use of IEQ>0.

Example: Note only one of the MESH namelist's is read, that specified by IMESH.

```
&STGF IMESH=1 PERT='YES' LRGLAM=8 IPRINT=-1 &END
&MESH1 MXE=600 E0=0.1193 EINCR=0.0005 &END
```

```
&MESH2 DQN=0.001 QNMAX=6.0 EMIN=0.1193 EMAX=0.2331 DEOPEN=0.0 &END
&MESH3 MXE=5 &END
0.45 0.50 0.60 0.80 1.00
```

It is IMPORTANT that you test that the resonances are sufficiently well resolved and that any that are not resolved are eliminated!

oooooooooooooooooooo

STGICF: Term-couple (unphysical) LS-coupling K-matrices to form intermediate  
 \*\*\*\*\* coupled collision strengths, following a STGF run.

Local executable: stgicf.x

STGICF Input

-----

The first line is used for user comments, then

Namelist STGIC

IMODE = 0 generate (write) jK or ic data (K-matrices and channel info) on the original (coarse) STGF energy mesh. Reads unphysical LS K-matrices generated by STGF run with IQDT=200 and IPRKM=4, following a normal STG1/STG2/STG3 run with RELOP='MVD'. Default.  
 = 1 read previously generated (IMODE=0) jK/ic data and calculate collision strengths on a fine energy mesh (input via namelist /MESH1/ as STGF).  
 =-1 single pass operation combining IMODE=0 and 1 without any disk I/O other than reading the STGF generated datafiles and writing to OMEGA.

ITCC = 0 pure jK-coupling.

= 1 intermediate coupling (default) requires  
 TCCDW.DAT from STG1/STG2/STGJK run with RELOP='TCC' and IDWOUT=2.

IBIGE = 0 default, does nothing.

= 1 read LS dipole line strengths (generated by STGF with IBIGE=1) to append infinite energy scaled collision strength to the file OMEGA.

INOEXCH = 0 default, does nothing.

= 1 non-exchange run, supplementing prior exchange run. Use IMODE=0, since no resonances present, only the coarse energy mesh is needed. Expects JMNTWO to be set, discussed next.

JMNTWO = twice the minimum value of J (total) to be included in the non-exchange run, following on from the last J of the exchange run.

= -1 default, but must be set if INOEXCH=1.

LRGLAM = -1 default, does nothing.

>= 0 applies Burgess dipole top-up at  $2*J=2*LRGL2=LRGLAM$ .

Normally, just set LRGLAM=999 (say) to top-up as code resets it to the largest appropriate  $2*J$ -value available.

In addition, the non-dipole transitions are topped-up assuming a geometric series.

Example: Single (stgicf) pass in the resonance region.

STGF data

```
&STGF IQDT=2 IMESH=1 PERT='YES' IPRKM=4 &END
&MESH1 MXE=101 E0=.01 EINCR=.00025 &END
```

STGICF data

C This line is for comments

```
&STGIC ITCC=1 IMODE=-1 INOEXCH=0 &END
&MESH1 MXE=2001 E0=0.01 EINCR=0.0000125 &END
```

See also the file stgicfadas.txt which gives a step-by-step guide to the practical usage of stgicf.

oooooooooooooooooooooooooooooooooooo

BOUND

\*\*\*\*\*

Read the H.DAT file generated by STG3 and calculate bound-state eigen-energies and eigen-vectors for the (N+1)-electron ion.

oooooooooooooooooooooooooooo

STGB: Seaton's perturbative treatment of the long-range coupling potentials.

\*\*\*\* Now extended to handle (electron+) neutrals.

Local executable stgb.x

STGB Input

-----

Namelist STGB

IPRINT defines the print level, 0 is the lowest and 3 the highest.

IPERT = 0 (default) Neglect long-range coupling potentials.

= 1 Include long-range coupling potentials, not normally necessary.

IOPT2 = 1 (default) Scan for bound states.

= 2 Read estimates of bound-state energy.

IRAD = 0 (default) Bound state energies only.

= 1 Write e-vectors to Bnm files for photoionization.

Then read (2S+1) L p or 0 2J p and scan range and step in terms of effective quantum number or energy estimates, depending on IOPT2. Terminate with -1 -1 -1.

Example: Fe25+

```
&STGB IOPT2=1 IRAD=1 IPERT=0 IPRINT=0 &END
0 0 1
0.5 5.5 0.025
0 0 0
0.5 5.5 0.025
0 2 1
0.5 5.5 0.025
0 2 0
0.5 5.5 0.025
0 4 1
0.5 5.5 0.025
0 4 0
0.5 5.5 0.025
-1 -1 -1
```

oooooooooooooooooooooooooooooooooooo

BOUND-FREE

\*\*\*\*\*

Read the Dnm files generated by STG3, Bnm by STGB and Fnm by STGF and then calculate photoionization cross sections. Run PREBF first then STGBF.

oooooooooooooooooooooooooooo

STGBF: Seaton's perturbative treatment of the long-range coupling potentials.  
\*\*\*\*\* Now extended to handle (electron+) neutrals.

Local executables prebf.x, stgbf.x

PREBF Input

-----

Namelist PREBF

IPRINT = 0 Default.

1 For more detailed printout

IBUT = 0 Default, no Buttle correction to orbital.

1 Orbital Buttle correction included.

WARNING: This simple input relies on the end-of-file to terminate gracefully so do NOT leave or insert any extra lines of data besides this NAMELIST PREBF else an attempt to read it will result in almost certain failure (if you are lucky, if unlucky.....)

STGBF Input

-----

Namelist STGBF

IPRINT=-2 Cross section for leaving the residual

ion in each of the target states, use for DR/RR.  
 -1 No printing of cross sections, levels  
 above ionization limit skipped. Used for production runs,  
 0 Total cross sections to a final SLP (default).  
 1 Files opened, qdt onset.  
 2 Partial cross section to each channel.  
 3 Beta and x-sectn to each target state.

IBUT = 0 Default, no Buttle correction to orbital.  
 1 Orbital Buttle correction included.

If STGF has been run in MQDT mode (IQDT=1 or 2) then also

MXE, E0, EINCR - meaning as per STGF - but now to define a new fine mesh.

EWIDTH > 0.0 Preconvolute TOTAL photoionization/recombination data with  
 this (z-scaled) width.

WARNING: This simple input relies on the end-of-file to terminate gracefully  
 so do NOT leave or insert any extra lines of data besides this  
 NAMELIST STGBF else an attempt to read it will result in almost  
 certain failure (if you are lucky, if unlucky.....)

oooooooooooooooooooooooooooooooooooo

#### BOUND-BOUND

\*\*\*\*\*

Read the Dnm files generated by STG3 and Bnm by STGB and then calculate  
 radiative data.

oooooooooooooooooooooooooooo

STGBB: For positive ions only, Seaton's perturbative treatment of the long-range  
 \*\*\*\*\* coupling potentials.

Local executables stgbb.x

STGBB Input

-----

Namelist STGBB

IPRINT = -1 Default, write f-values to file FVALUE,  
 1 and write line strengths to UNIT6 as well,  
 0 and write f-values to UNIT6 as well.

IBUT = 0 Default, no Buttle correction to orbital.  
 1 Orbital Buttle correction included.

Then read pairs of terms (2S+1) L p or levels 0 2J p for which radiative  
 data is required. Terminate with -1 -1 -1 -1 -1 -1, or end-of-file.

oooooooooooooooooooooooooooooooooooo

#### RADIATION + AUGER DAMPING

\*\*\*\*\*

Read the H.DAT and Dnm files generated by STG3 and Bnm by STGB.

oooooooooooooooooooooooooooo

STGFDAMP: A radiation+Auger damped version of STGF. Note, Type-I damping only  
 \*\*\*\*\* can be obtained from STGF with IRDEC=1 (and NDRMET>0 for OMEGDR).

Local executable stgfdamp.x

STGFDAMP Input

-----

Namelist STGF

All variables present in STGF plus switches to turn radiation+Auger damping on (=1) or off (=0).

NTYP1 = 1 (Core electron) radiation damping to non-STGB states, default.

NTYP2I = 1 In-the-box damping to STGB states, default.  
Requires dipole matrices to these states as in a normal (undamped) photoionization calculation with STGBF i.e. RAD='YES' in stg1/2/3

NTYP2OR = 1 Outer electron radiative stabilization to non-STGB states, default.

NTYP2OF = 1 Radiative recombination to non-STGB states.  
=-1 Not set. Default, since NDRMET=0 by default.

NMIN = lowest principal quantum number n to apply NTYP2OR/F.  
Default=-1, not set. Attempts to determine it from B00 but if NTYP2I=0 (and NTYP2OR/F=1) it must be set by the user.  
It should be 1 greater than the largest stgb n-value.

In non-MQDT mode, sufficient recombined states must be generated via STGB so that the in-the-box damping of the omitted states is negligible. Outer region damping is done via a complex energy and theta function.

In MQDT mode, only those final states formed by bound orbitals input from STG1 should be generated via STGB - they have been projected out of the continuum basis. All other final states are taken into account (both inner and outer region damping) via a complex energy in MQDT.

NDRMET >0 Determine DR from the first NDRMET metastables, output to OMEGDR.  
Note: this switches on elastic transitions and which are now present in the OMEGA file.  
=0 None (default).

oooooooooooooooooooo

STGICFDAMP: Term-couple (unphysical) LS-coupling damped K/S-matrices to form  
\*\*\*\*\* intermediate coupled collision strengths, following a STGFDAMP run.

Local executable: stgicfdamp.x

STGICFDAMP Input

-----

The first line is used for user comments, then

Namelist STGIC

All input variables from stgicf plus:

NTYP1 = 1 Include Type-1 (inner electron) damping (default). Note, Type-2 (outer-electron) damping would have been included back in the stgfdamp run and is built into the now complex K-matrix.

NDRMET > 0 Determine DR from the first NDRMET metastables, output to OMEGDR.  
Note: this switches on elastic transitions and which are now present in the OMEGA file.

= 0 None (default).

IQDT = 0 Looks for a K-matrix file from a stgfdamp run. If not found it looks for an S-matrix file instead.  
 = 1 Look for S-matrix file smtls.dat only and stop if not found.  
 = 2 Look for K-matrix file z/kmtls.dat only and stop if not found.

Note extension to INOEXCH variable:

INOEXCH > 0 Read complex K-matrices on zkmtls.dat from a STGFDAMP NX run.  
 < 0 Read real K-matrices on kmtls.dat from a STGF NX run.  
 = 0 Read complex K-matrices on zkmtls.dat from a STGFDAMP exchange run, default.

oooooooooooooooooooo

STGBF0DAMP: A radiation+Auger damped photoionization code - initial  
 \*\*\*\*\* states MUST be fully contained within the R-matrix box.

Local executable stgbf0damp.x

STGBF0DAMP Input  
 -----

Namelist STGF

All variables present in STGFDAMP plus switches to specify the initial states to photionize from.

IPHOTO = Number of electron continua to photoionize to. Default = 1.  
 (Zero gives no photoionization.)

NPISYM = Number of STGB symmetries to photoionize from. Default = 1.  
 (Taken in the order they specified in the STGB run.)

NPIEB = Number of STGB energies per symmetry to photoionize from. Default=1.  
 (Taken in the order they were determined in the STGB run.)

oooooooooooooooooooooooooooooooooooo

Utility Codes  
 \*\*\*\*\*

ADASEX: Produce ADAS adf04 file (LS-resolved). Minimal input - see code.  
 \*\*\*\*\*

ADASEXJ: Produce ADAS adf04 file (IC-resolved). Minimal input - see code.  
 \*\*\*\*\*

\*\*2ASCI and ASCI2\*\* : Programs to convert files from binary to ASCII and  
 \*\*\*\*\* vice versa. Not the most elegant way to port between  
 machines, but it works...

OMADD: Add two OMEGA files together. Typically a fine meshed exchange one  
 \*\*\*\*\* including resonances and a coarse meshed non-exchange. The latter is  
 interpolated onto the former. In addition, it can analyse resonance  
 resolution and eliminate unresolved resonances. Can also be used to  
 compare two OMEGA files - one with top-up and one without and produce  
 detailed diagnostics on the top-up contribution, both dipole and  
 non-dipole. Default operation (adding) requires no input. See code  
 for options for other operations.



stg2r.x) reads target eigen-energies and eigen-vectors from a file MIXDW.DAT and collision algebra from a file ALGDWU.DAT, both written by STG2 of the EXCHANGE codes, FOR ALL L. It then evaluates continuum distorted-waves and continuum-continuum (reactance) matrix elements and writes collision strengths in the standard form to the file OMEGDW. Note, for large cases the local executable stg2rdw.x should be used in place of stg2r.x.

#### STG1 Input

-----

In namelist STG1A, specify

IDWOUT= 0 (default) for CC only.  
= 1 for CC and DW.  
= 2 for DW only, recommended (fast).

All other variables are as before (see 1.). However, if IDWOUT=2, only MAXLA needs to be specified and the other variables are ignored.

Example: N,

S.S.

```
&STG1A IDWOUT=2 &END  
&STG1B MAXLA=4 &END
```

#### STG2 Input

-----

In namelist STG2A, IDWOUT, same meaning as above, is specified along with ISORT which is discussed below.

In namelist STG2B, in addition to the variables described above (see 1.) for STG2 input, NMETA and (optionally) IMETA(N) are specified where,

NMETA is the number of (initial) metastables (including the ground) from which collision data is required. If

ISORT= 0 the first NMETA target terms listed (see 1.) are the metastables. This is only useful if the terms are energy-ordered, which is inefficient for large cases.  
=-1 the NMETA metastables are defined by IMETA(N)=1 where N is the position in the input term list, which maybe symmetry or energy-ordered.  
= 1 the term list is sorted into symmetry-order and the lowest energy term of the first NMETA symmetry groups is taken to be metastable. This can be overridden by IMETA(N)=1 where N is the term position in the SORTED list. HOWEVER, the term list is expanded to explicitly include all correlation terms of the same symmetry that were in the original term list.

The specification of the N-electron configurations and term list follows as before (see 1.).

BUT, there is NO specification of (N+1)-electrons configurations since the Distorted-Wave code uses non-orthogonal continuum orbitals and evaluates exchange-overlap integrals instead. This is important if you use INAST>0 since the (N+1)-electron SLp symmetries now follow directly the target terms. If INAST=0 then nothing is read after the target terms and the above discussion can be ignored.

LNOEX = the highest L for which all exchange multipoles are calculated.  
Default = 10.

Example: (partial) N,

S.S.



```

&STG2A IDWOUT=2 ISORT=-1 &END
&STG2B MAXORB=6 NELC=7 NAST=34 NMETA=3 IMETA(1)=1 IMETA(3)=1 IMETA(6)=1
      INAST=0 MINLT=0 MAXLT=8 MINST=1 MAXST=5 &END
1 0 2 0 2 1 3 0 3 1 3 2
6
2 0 2 0 0 0
2 2 5 1 1 1
2 2 3 0 0 0 0
2 1 4 0 0 0 0
2 0 5 0 0 0 0
2 2 2 1 0 0 0
2 2 2 0 1 0 0
2 2 2 0 0 1 0
4 0 1
4 0 1
2 2 1
2 2 1
2 2 1
2 1 1
2 1 1
2 1 1
2 1 1
2 1 1
4 1 0
4 1 0
4 1 0
2 1 0
2 1 0
2 1 0
2 1 0
2 0 1
.
.
.
.
etc.

```

## STGDW Input

-----

In namelist STGDW, specify

INAST = the number of SLp symmetries. The first INAST symmetries generated in STG2 are included, see also LTOP and LCUT below for alternative ways to read a subset of the collision algebra.  
Default = 999, all symmetries present.

MXE = the number of user supplied UNSCALED energies in Rydbergs relative to the ground state which are read free-formatted after this namelist.  
MUST be specified > 0.

NMETA <= the number of metastables specified in STG2, the first NMETA STG2 metastables are used.  
Default = the value of NMETA specified in STG2.

NAST = Number of observed energies (Ry) to be read AFTER the MXE scattering energies, relative to the ground and in the same order as listed in the STG2 data (or in the MIXDW.DAT file). Default = 0, none.

ELAS='NO' (default) no elastic data calculated.  
='YES', elastic collision strengths calculated and written to file OMEGDW.

IPERT = 0 neglect contribution from long-range integrals outside of the atom - similar in spirit to PERT in the STGF outer-region code.  
= 1 (default) include full long-range contribution. See LRANGE next:

LRANGE = 0 (default) evaluate long-range integrals (if IPERT=1) to the end of the continuum mesh.

= 1 Add-in the contribution from beyond the end of the mesh, particularly slow for a small mesh, otherwise just slow.

IPRTM defines the print level, -1 the lowest and +1 the highest. Default=0.

LTOP = Highest total L for which explicit data is to be used. The contribution from LTOP+1 to Infinity is then calculated assuming a geometric series in L.

Default LTOP=999, no "top-up".

LCUT = Exclude contribution from L>LCUT. Only useful if "top-up" (LTOP) is not used since specification of LTOP imposes an LCUT in practice.

Default = 999, no cut.

NTHRSH = 0 (default) use only the input MXE energies.

= 1 calculate every threshold collision strength. This can be useful for ions since we tend to use a very coarse energy mesh, the collision strengths can be splined subsequently.

DO NOT USE FOR NEUTRALS since you know already that the collision strength is zero then.

LNOEX = Highest L for which exchange is included (if indeed present). Much better to use LNOEX in STG2 to save calculating and writing it in the first place.

Default = 999, all exchange present included.

IDFOT = 0 Default.

= 1 Reserved output for differential code not discussed here.

= 2 Reserved output for differential code not discussed here.

= 3 Write reactance (K-) matrix elements to 'rmtls' for STGDWIC.

NUNIT = 0 Default. Non-unitarized. Fast.

= 1 Unitarized. requires NMETA=NAST in STG2. Slow.

Example: N,

```
&STGDW INAST=54 MXE=5 NTHRSH=0 NMETA=3 ELAS='YES' IPERT=0 IPRTM=-1 LTOP=8 &END
1.0
1.5
2.0
3.0
4.0
```

Note, INAST=54 corresponds to the number of symmetries generated by the STG2 specification MINLT=0 MAXLT=8 MINST=1 MAXST=5 (both parities).

The same results could have been obtained with a reduced namelist viz.

```
&STGDW MXE=5 ELAS='YES' LTOP=8 &END
```

oooooooooooooooooooooooooooooooo

IC

\*\*

The IC distorted-wave code (local executable stgdwic.x) again reads MIXDW.DAT as well as term coupling coefficients from TCCDW.DAT and reactance matrix elements, newly generated by STGDW, from the file 'RMTLS.DAT'.

The TCC's are generated most efficiently for a DW-only run by setting IDWOUT=2 and RELOP='TCC' when running through STG1, STG2 and STGJK, see 2.

Note, this is a separate run from that which generated the LS collision algebra.

